ARMY RESEARCH LABORATORY

An Automated Tool for the Creation of Desired Emulated Topologies of a Mobile Ad-Hoc Network

by Binh Q. Nguyen

ARL-TR-3721 January 2006

NOTICES

Disclaimers

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-3721 **January 2006**

An Automated Tool for the Creation of Desired Emulated Topologies of a Mobile Ad-Hoc Network

Binh Q. Nguyen
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
January 2006	Final	01 Oct 04 – 30 Sep 05

4. TITLE AND SUBTITLE	
An Automated Tool for the Creation of Desired Emulated Topologies of a Mobile Ad-Hoc Network	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Binh Q. Nguyen	**5d. PROJECT NUMBER** 5FE6BG
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-CN 2800 Powder Mill Road Adelphi, MD 20783-1197	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-3721
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory 2800 Powder Mill Road Adelphi, MD 20783-1197	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report documents a research effort that the U.S. Army Research Laboratory conducted in FY05 in support of the development of advanced technologies for tactical mobile ad hoc networks (MANET). The effort resulted in the creation of a visually appealing tool called *TDFCreator*. The purpose of the tool is to provide researchers with an automated tool and methods for creating desired topologies and generating topology scenario files through visual manipulation, inspection, and animation of changing topologies. The tool is easy to use as it has a graphical user interface and uses a computer pointing device with which the user is well familiar. The tool benefits its users in (1) providing several methods for improving design accuracy through visualization of newly created topological links among participating nodes using, (2) automatically managing the size and the complexity associated with the creation of topology files, (3) significantly reducing the time required to design and develop specific topologies required for the emulation of a MANET, and (4) avoiding the tedium of manually writing topology directives, effectively enabling its users to focus their efforts on the design of topologies rather than on the manual generation, manipulation, and inspection of topology directives.

15. SUBJECT TERMS

Mobile ad hoc network, desired topology, emulation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Binh Q. Nguyen
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UL	40	**19b. TELEPHONE NUMBER** (*Include area code*) 301-394-1781

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Preface

This work was performed in support of the building of a tactical environment assurance laboratory (TEAlab) at the U.S. Army Research Laboratory, Adelphi, Maryland. The TEAlab is an emulated mobile ad hoc network (MANET) research test bed environment in which information-assurance technological products and novel routing protocols are developed, tested, evaluated, and demonstrated in collaboration with industry and academia research partners.

This work came to fruition with a useful and appealing tool called *TDFCreator* that provides researchers methods for graphically designing desired MANET topologies and for automatically generating topology definition files. The tool provides a method for generating position locations associated with the topology in support of visualization and animation.

Executive Summary

A mobile ad hoc network (MANET) topology is a time-dependent snapshot giving the picture of the communication links among the participating nodes. The emulation of dynamic topologies can be accomplished using a fixed local area network and a packet filtering software. The emulation of MANET topologies at the U.S. Army Research Laboratory (ARL) uses a tool developed by Telcordia Technologies, Inc., called "topology scenario manager" (TSM).

The input of the TSM tool is a file consisting of sequences of directives defining the topology of a MANET under test; hence, the input file is called a topology definition file, commonly called a TDF file. The directives are written in the scenario definition language (SDL), also developed and used by Telcordia Technologies, Inc., specifying appropriate link commands and the time duration of each topology. The time duration controls the speed of the changing topologies. The TSM interprets the directives and translates them into *iptables* commands that effectively control the topology.

The manual creation of TDF files is a complex, error-prone, time-consumptive, and monotonous process. The number of directives in a TDF file is a quadratic function that depends on the size of a MANET and the number of snapshots preferred by a researcher. To accelerate this process, ARL has independently and successfully designed, developed, and used an easy-to-use tool called *TDFCreator* capable of assisting researchers in the creation of desired MANET topologies and of automatically generating and saving the directives in TDF files. The TDF files are subsequently combined to create dynamic topologies of a moving MANET.

The *TDFCreator* eases the learning curve because it incorporates graphical user interfaces and a computer pointing device familiar to users. The pointing device can be either a computer mouse or a touchpad or a track point that is often built into a notebook computer. The left mouse button is used to create a new node or a new link between two nodes. The right mouse button is used to delete an existing node or an established link. Therefore, the learning curve is substantially minimized as a researcher can learn how to use the tool within minutes.

The *TDFCreator* benefits the researcher in at least four different ways. First, the tool provides several methods for improving design accuracy through visualization of simultaneous topological links among nodes. Second, the tool significantly reduces the time required to design and develop specific MANET topologies and implementing them using the SDL because the tool automatically generates TDF files. Third, the tool assists the user in managing the size and the complexity associated with the creation of TDF files, effectively and expeditiously. Fourth, the tool exempts its users from the direct use of the scenario definition language, in effect redirecting their focused efforts on the automated design of topologies rather than the manual writing of topology directives.

In conclusion, the *TDFCreator* is a useful tool capable of notably increasing the productivity of its users. It is being used not only within the ARL, but also at Telcordia Technologies, Inc.,—a premier research institution and a Collaborative Technology Alliances member of the ARL. Using the *TDFCreator*, researchers have been, for the first time, systematically and visually designing topologies and creating TDF files quickly and reliably. By using the *TDFCreator*, its users can save time, enhance productivity, reduce complexity, and avert errors. This development has provided researchers with an unprecedented tool for creating desired MANET topologies and for effectively managing the volume and the intricacy associated with TDF files.

1. Introduction

1.1 Ad Hoc Networks

An ad hoc network system generally consists of a wireless network of cooperative devices called nodes capable of communicating with one another directly or indirectly without relying on any centralized services or a previously established infrastructure. Direct links between any two nodes occur when one node can successfully receive a message sent by the other node. When any two linked nodes can mutually receive messages from one another, the links are bidirectional; otherwise, the links are unidirectional. Indirect links between two nodes rely on other intermediary nodes that act as network routers, automatically forwarding messages based on their destination addresses using a routing table. All nodes can act as network routers. The communicating nodes and their connections at any given time form a particular network topology, which is defined by the Microsoft Network Encarta dictionary as “the relationships between parts linked together in a system such as a computer network.”

An ad hoc network can be mobile or stationary. Stationary ad hoc networks usually consist of environmental sensors (e.g., acoustic, temperature, seismic, motion, infrared, and magnetic sensors). Once deployed, these sensors stay in the same place; therefore, their topology remains fixed unless a participating sensor stops communicating. Whereas, a mobile ad hoc network (MANET) is an autonomous system of mobile nodes (*1*). Each node is a networked computing device capable of receiving and transmitting data using radio frequencies (RF). A MANET could be comprised of portable communicating devices being used in geographical areas in which an information infrastructure does not exist or has been damaged or destroyed (due to sabotage, terrorism, war, or natural disaster). The topology of a MANET dynamically changes as the nodes move from one location to another.

The U.S. Army Research Laboratory (ARL) and its Collaborative Technology Alliances (CTA) partners are developing MANET technologies to provide future mobile tactical forces of the U.S. Army with advanced wireless communications network systems.

1.2 Emulated Mobile Ad Hoc Networks

Conducting research in MANE) requires the establishment of an appropriate network environment in which relevant technologies can be effectively developed, tested, evaluated, and demonstrated. Implementing a physical environment requires substantial resources in terms of personnel, time, money, and space. Creating a simulated MANET model is a lower-cost alternative, but running actual protocols and applications in the model is ineffective if not possibly infeasible. Building an emulated MANET provides a realistic environment with lower costs and enables repeatable execution and evaluation of actual MANET protocols and security applications in physical environments.

Existing techniques used to emulate topologies of a MANET include the use of a system software entity to control the flow of data traffic seen at individual hosts or at the central network switch to which all communicating nodes are connected. The latter technique requires the establishment of a *wired* local area network of computers connected to a central, dedicated computer equipped with many network interface ports, such as the Dynamic Topology Switch (DTS) (*2*). Whereas, the former technique does not require any special networking technology or a special setup; it can be implemented on an ordinary wired or wireless local area network using a packet filtering software; e.g., *iptables*. ARL and its partners have adopted the former technique, as it is simpler and more flexible.

The dynamic execution of *iptables* commands controls the flow of network data among participating networked computers in real time, effectively enabling stationary computers to see themselves as if they were moving and causing topological changes. Sequences of timed *iptables* commands are issued and executed to emulate a dynamic MANET scenario. However, for emulating a large and meaningful MANET, creating these *iptables* commands for direct execution is not practical because the commands are highly cryptic.

To avoid dealing directly with the low-level *iptables* commands, the ARL and its partners have adopted the scenario definition language (SDL) to design emulated MANET topologies. Using SDL, researchers can create desired topologies by writing sequences of human-readable topology directives and save them in a text file. This file is called a topology definition file or simply a TDF file. The file contents are subsequently interpreted and translated into *iptables* commands by a topology scenario manager (TSM).

The TSM is a proprietary tool of Telcordia Technologies, Inc., which was a division of Science Applications International Corporation (SAIC) when it licensed the TSM to the ARL through the CTA program of ARL in 2003. The SDL is also used in the SAIC-developed military communications scenario generator (CSG) to textually express the visibility between any two communication entities (*3*).

1.3 Challenges in the Creation of Topology Definition Files

The emulation a MANET requires the creation of TDF files. The manual creation of TDF files using SDL is an improvement over using the *iptables* commands directly, but still an error-prone, time-consumptive, and monotonous process. The number of directives in a TDF file is a quadratic function, $f(m,n)$, that depends on the size of the MANET (n) under test and the number of network topologies (m) chosen by a researcher.

To create a static topology for an *n*-node MANET, a topology designer must decide and implement n-1 topological directives for each node using an ordinary text editor; e.g., *notepad*, *vi*, or *emacs*. Each directive is a line of SDL code specifying the action to be performed on an inbound data packet, either to deny or to accept it. Therefore, the number of lines of code, p, is a function of the number of participating nodes n, i.e., $p(\mathrm{n}) = n(n–1)$. To emulate the mobility of this network, the scenario needs m distinct topologies, interspersed with roughly m·n lines of time-duration commands. The total number of lines of SDL code, T, requires $m{\cdot}n(n–1)$ topological directives and $m{\cdot}n$ interspersed time duration commands, i.e., $T = f(m,n) = m{\cdot}n(n–1) + m{\cdot}n = mn^2$. For example, to run a ten-minute emulation of a 30-node MANET whose location is updated every second, the number of lines of SDL directives in the associated TDF file can be up to 540,000 (10 min/sec x 60 sec/min x 30 x 30).

In summary, the manual process of creating topological directives is complex, time-consumptive, error-prone, and tedious. The size and the complexity of a TDF file containing sequential instructions for emulating MANET dynamic topologies pose a monumental challenge to topology designers. Textual descriptions of topological specifications further compound to the problem because they are difficult to comprehend and to keep track of changes. They can also contain typographical and sometimes logical errors or unintended instructions associated with the use of a text editor to create TDF files.

Because of these challenges, for many years under the current CTA program and its predecessor, the Federated Laboratory (FedLab) program, researchers at the ARL and at Telcordia Technologies, Inc., could emulate only a small-sized MANET consisting of only a few nodes.

Developing, evaluating, and demonstrating advanced MANET routing protocols and security products on a small-sized MANET will undoubtedly raise the issues of whether the results would be valid, credible, realistic, and reliable. Therefore, emulating a large MANET was necessary to overcome these potential issues. However, creating emulated topologies for a large-sized MANET remained a great challenge until the successful creation of an automated tool called *TDFCreator*.

1.4 Purpose and Scope of the Report

This report documents an unprecedented software tool called *TDFCreator* that has been independently designed and developed by the ARL. The tool is being used by internal and external researchers to visually create desired topologies and to automatically generate TDF files for the emulation of a tactical MANET. This report has the following main purposes:

- Reporting the results of the internal development of the *TDFCreator* tool that the Battlefield Communication Networks Branch has contributed to the mission of the ARL as of the end of the 2005 fiscal year.
- Documenting an ARL contribution to the field of networked communications, especially stationary and MANETs, and principally to a method for visually designing and verifying desired network topologies and automatically creating topology definition files.
- Describing the features, benefits, and capabilities of the *TDFCreator* tool as of September 2005.
- Presenting potential improvements to the *TDFCreator* tool in the future.

The intended readers of this report include practicing MANET researchers, engineers, and technical managers.

The next section includes several topics related to the emulation of a MANET using an ordinary network. It briefly describes the emulation process at the ARL, related emulation tools, how the emulation is being accomplished at ARL, and the role of the *TDFCreator* in the process. Section 3 explains the *TDFCreator* tool, consisting of its design philosophy, provided methods, graphical user interfaces, and output files. Section 4 presents and discusses the results of the use of the tool and a plan for improving it. Section 5 concludes the report by summarizing and highlighting the benefits of the tool.

2. Emulation Process at the ARL

As mentioned in the previous section, emulation that uses *iptables* commands to create a desired logical topology is a technique that has been widely used in laboratory environments. The *iptables* commands are input directives for the *iptables* packet-filtering system that controls the inbound data packets in real time to emulate the connection between any two communicating computers. The packet-filtering process is based on the hardware address of the network interface card of the sender, and it can be accomplished in a wired or wireless network.

2.1 Tools

Among the tools that the ARL studied, the SAIC-developed CSG tool (*3*) was the most relevant tool because it can generate topology definition files that the TSM can execute. The SCG tool provides a way to design and to animate military communications scenarios and to generate databases that can be used to feed other simulation tools.

The ARL initially had planned to use the CSG tool to generate topology definition files; however, after an empirical evaluation of the tool, the ARL concluded that using the tool in the early phase of the project would be untimely for the following reasons:

- The CSG-generated topology definition directives are based on the calculated line-of-sight (LOS) visibility between any two nodes, which depend on their locations on a digital elevation map or curved earth terrain. This feature thus does not provide researchers a convenient way to custom-build a desired topology. Furthermore, using digital elevation maps in the emulation of a dynamic MANET remains a plan at the ARL.

- The CSG tool is not well suited for emulating dynamic MANET scenarios, but rather for generating classic operational military communications scenarios involving conventional communications equipment.

- The CSG tool requires a steep learning curve as it is a very military-domain specific tool with which a civilian researcher is rarely familiar.

Other existing and related tools that also use *iptables* to enable the building of emulated MANET include the MobiEmu tool (*4*), the mobile network emulator (MNE) (*5*), the mNet tool (*6*), and MASSIVE (*7*). Not one of these tools uses the SDL that has been adopted and used by ARL and its partners to specify a MANET topology. All these tools take in different input control directives, and therefore, they are not interoperable although they generate similar *iptables* commands as the TSM tool used by the ARL does.

The ARL uses an eclectic set of tools to emulate a MANET test bed as depicted in figure 1. The emulation process consists of two phases. The first phase uses the internally developed *TDFCreator* tool to visually design desired topologies and to automatically generate scenario files called TDF files required for execution in the physical test bed. Leveraging the work of others, the second phase uses existing tools to analyze, visualize, and emulate a dynamic MANET in real time on an ordinary fixed network.

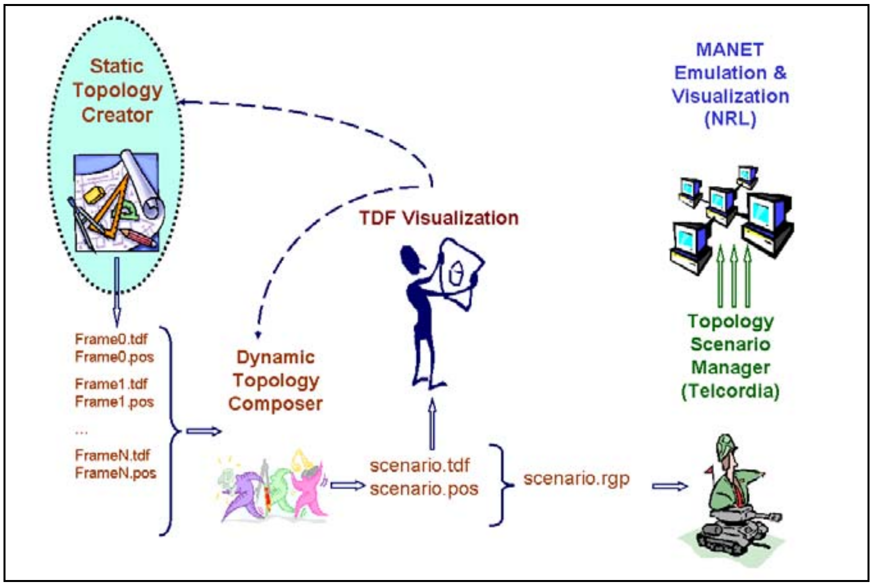

Figure 1. The role of the *TDFCreator* tool in the emulation of a MANET.

The existing tools include the TSM tool and the *Nettion* tool set. The U.S. Naval Research Laboratory developed the *Nettion* tool set consisting of data communications, analysis, and visualization tools (*8*). Telcordia Technologies, Inc., developed the TSM tool and delivered the version 1.2 of the tool to the ARL in September 2003. As this version could handle only the SDL commands, it was modified by ARL in early 2005 to enable it to handle the position information needed for real-time communications with a display tool called *JMAP*, which is part of the *Nettion* tool set.

As the ARL-modified TSM tool requires location information associated with each node, the TDF files had to be modified to incorporate the additional requirements. The new TDF files have the extension file type *rgp*, which stands for the three name initials of the ARL engineer who modified the TSM tool.

As the focus of this report is on the *TDFCreator* tool, detailed descriptions of the actual emulation of a MANET and the tools used in the process are beyond the scope of this document.

2.2 Topology Design and Development

The topology design and development phase, the first phase in an emulation process, requires the creation of desired topologies and topology definition files. This phase has two major sub-processes: (1) the visual design of static and dynamic topologies and their time durations and (2) the preview of animated sequences of the newly created topologies in the order they were saved. Once the design phase ends, the topology definition files are combined into a single file called the dynamic scenario file, readily for execution in each individual networked computer participating in the emulation of a MANET.

2.3 Topology Execution

The topology execution phase emulates a MANET. It is performed on a local area network (LAN) of computers running the Linux operating system. The enabling tool in this phase is the ARL-modified TSM tool that takes in the dynamic scenario file and performs the following main actions:

- Interpreting and sending the location information to the *JMAP* display tool for visualization and monitoring purposes
- Interpreting and translating the SDL directives into *iptables* commands to form a logical topology
- Suspending itself (*i.e., sleep*) for a number of seconds as specified in the SDL *wait* statement. The sleep time indicates the duration of the currently emulated topology.

3. The *TDFCreator* Tool

The *TDFCreator* tool relates to networked communications, especially stationary and MANETs, and principally to a method for visually designing and verifying desired network topologies and automatically creating topology definition files. Learning how to use the tool is easy because the tool has a graphical user interface and uses a computer pointing device with which the user is well familiar.

3.1 Purposes and Uses

The *TDFCreator* tool has been independently devised and implemented by the ARL to solve the challenges posed by the manual design process. The primary purpose of the tool is to provide researchers with an automated tool and methods for creating desired emulated topologies through visual manipulation, verification, and animation of changing topologies.

The *TDFCreator* tool enables the user to use a pointing device to simply point, click, and drag to create and to place graphical elements representing a desired topology and to automatically generate the associated topology definition files needed for the emulation of a MANET. The *TDFCreator* tool automatically interprets and translates the graphical presentation of a topology into textual descriptions expressed in SDL.

3.2 Provided Methods

The *TDFCreator* tool provides the following methods for creating emulated topology definition files that are necessary for the emulation of a MANET:

(a) A method for graphically designing desired topologies by placing, relocating, or removing graphical elements representing MANET nodes and the links among the nodes on a screen instead of manually laying out and sketching them on paper and then writing their descriptions as sequences of SDL commands using a text editor.

(b) A method for improving design accuracy through visual inspection and animation of newly created topological links among participating MANET nodes using a graphical tool instead of manually reading through sequences of textual descriptions of topology definition.

(c) A method for automatically managing the size and the complexity associated with the creation of topology definition files using effective data structures and algorithms. A MANET is implemented as a dynamic list of mobile nodes. Each node is an object that keeps track of its position locations on the screen and uses a dynamic list to store the information of its connections to other nodes. No human intervention is necessary.

(d) A method for automatically generating topology definition files through automatic interpretation and translation of graphical elements representing a topology into sequences of topology definition directives. Once again, no human intervention is necessary.

(e) A method for visually conducting dry runs by previewing sequences of changing topologies through automatic composition and animation of saved topology files.

(f) A method for significantly reducing the time and the effort required to design specific MANET topologies and to implement them using SDL.

(g) A method for avoiding the tedium of manually writing sequences of topology definition directives to express the implementation of desired topologies and saving them to permanent storage files because the tool provides an automated method for generating the files.

(h) A method for averting typographical errors often associated with the manual writing of topology definition directives to express the implementation of desired topologies because the tool provides an automated method for generating the files.

(i) A method for focusing efforts on the design of topologies rather than on the manual generation, manipulation, and inspection of topology definition directives as the tool is an automated tool.

(j) A method for dealing with the emulation of a large-sized MANET via graphical user interface and automatic generation of topology definition files.

(k) A method for graphically illustrating the behavioral complexity of a MANET to unfamiliar users for the purpose of education and training.

(l) A method for graphically demonstrating to actual and potential customers and supporters the capability of the test bed and the emulation of a MANET.

(m) A method for increasing the productivity of researchers through automation and ease-of-use.

3.3 Design Philosophy

The *TDFCreator* tool was created with a set of principles that form the basis and the requirements for its existence. The principles include simplicity, usability, portability, reusability, compatibility, and adaptability. To realize these principles, the ARL has adopted the Python programming language and its standard graphical user interface toolkit called Tkinter (*8*). Furthermore, as the Python syntax was simple and thus easy to learn, it was used to expedite the development process in order to meet the pressing needs of the ARL for having an emulated MANET test bed in no time.

3.3.1 Simplicity

Plainly functional is a main characteristic of the tool. It does what it is designed to do: provide the user with a graphical tool for visually custom-designing desired topologies and automatically creating scenario files. Every feature of the tool serves a useful purpose; not a single feature of the tool was designed to dazzle or to impress an audience.

3.3.2 Usability

With the focus on its users, the ARL designed and developed the *TDFCreator* tool to provide an effective and efficient tool that can be easily used by a MANET researcher to quickly create emulated topology definition files with minimum effort. The tool incorporates graphical user interfaces and a computer pointing device with which the user is well familiar.

Graphical interfaces include a drawing canvas, buttons, text boxes, and radio buttons as shown in figures 2 and 3. The background image or color and the dimensions of the canvas are pre-configurable to fit the preference of an individual user or to meet the unique requirements of an emulation scenario. Action verbs expressed in English text for quick comprehension without memorization are used to label the buttons. Text boxes are provided for the user to enter textual inputs such as file names and duration times. Radio buttons are used for selecting an input state of the tool. Only familiar iconographical representations are used in tool to ensure easy recognition. For examples, the two ARL logos are always displayed on the upper left and on the upper right of the screen to identify the origin and the ownership of the tool. These two logos cannot be removed by the end-user.

The pointing device can be either a computer mouse or a touchpad or a track point that is often built into a notebook computer. The left mouse button is used to create a new node or a new link between two nodes. The right mouse button is used to delete an existing node or an established link. Therefore, the learning curve is substantially minimized as the user can learn how to use the tool within minutes.

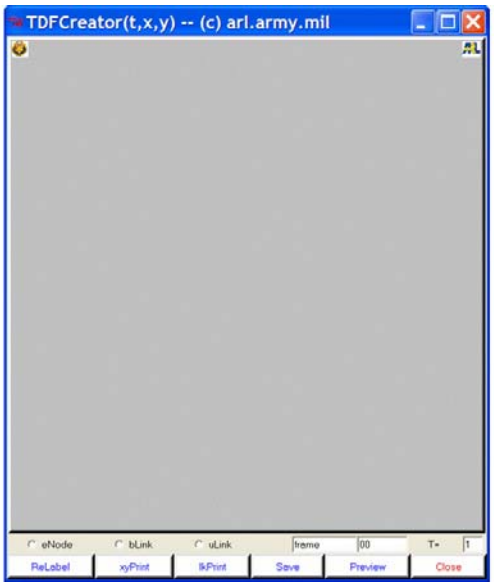

Figure 2. The *TDFCreator* tool and its gray canvas.

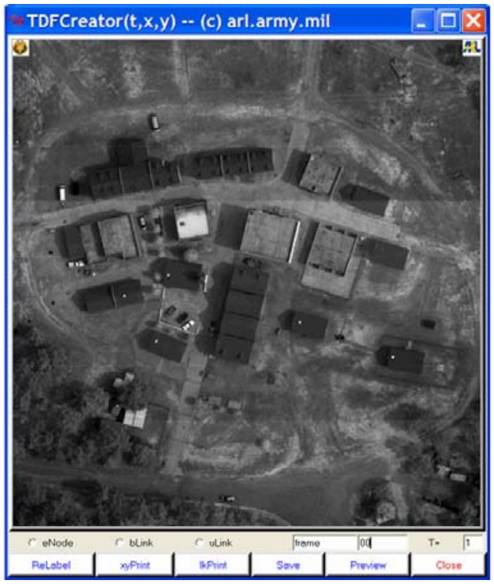

Figure 3. The *TDFCreator* tool running with a loaded background image.

3.3.3 Portability

The ability of the tool to run in dissimilar operating environments minimizes the time required to develop, install, configure, and run the tool. The tool extensively uses common system services that Microsoft Windows and Linux operating systems provide. Although the *TDFCreator* tool was developed and tested in the Microsoft Windows XP environment, it runs in the Linux operating systems without any special modification.

3.3.4 Reusability

As Python is an object-oriented language, previously implemented code can be reused, inherited to minimize the effort and the time required to add a new functionality. The *TDFCreator* tool reuses the canvas object that was also used in the visualization tool, a related tool that has been integrated into the *TDFCreator* tool for previewing purposes.

3.3.5 Compatibility

The *TDFCreator* tool generates topology scenario files that are compatible with the input formats of the two versions of the TSM program. The TSM program itself is an experimental computer program that has been modified several times. Each version of the TSM program requires a different input format. The original TSM program takes in only SDL commands; it does not process position location information. That was the main reason that the *TDFCreator* tool generates and saves the position information separately in the *pos* files. The ARL-modified TSM can take both SDL commands and position location information.

3.3.6 Adaptability

The *TDFCreator* tool is adaptable to changing requirements as it is implemented in the Python programming language. As the tool belongs to the ARL, ARL can modify or improve the tool whenever necessary to meet the new requirements without having to obtain any licenses from a third party. For example, the output of the tool can be changed to generate topology directives in any data format required by a different tool or a switching device that controls the emulation of a MANET. For example, should the DTS be used in the future, the *TDFCreator* tool can be easily modified to generate the "switch connectivity table" required by the DTS (*2*).

3.4 Input Methods

Input methods extensively use graphical user interface and a pointing device having at least two buttons (e.g., a computer mouse). The left button is used to create a new node or a new link between two nodes. The right button is used to delete an existing node or an established link. Figures 4 through 9 illustrate an example of the creation and modification of a fully connected star topology using a five-node MANET.

3.4.1 Node Representation

Figure 4 shows the successful creation of five participating MANET nodes. This process has two steps. First, the radio button labeled "eNode" is selected to start the creation of new nodes and the relocation or the deletion of a currently displayed node. Second, a new node is created by moving the cursor over the screen image, stopping it at a desired location, and pressing and releasing (single click) the left button. The deletion of a node is accomplished by simply moving the cursor over the target node and clicking the right button.

Figure 4. Creation of MANET nodes.

3.4.2 Unidirectional Links

Figure 5 shows the creation of unidirectional links between two MANET nodes. Messages sent from nodes 0, 1, 2, and 4 can be received by node 3, but messages sent by node 3 cannot reach any other node. A single-head arrow represents a unidirectional link between two nodes.

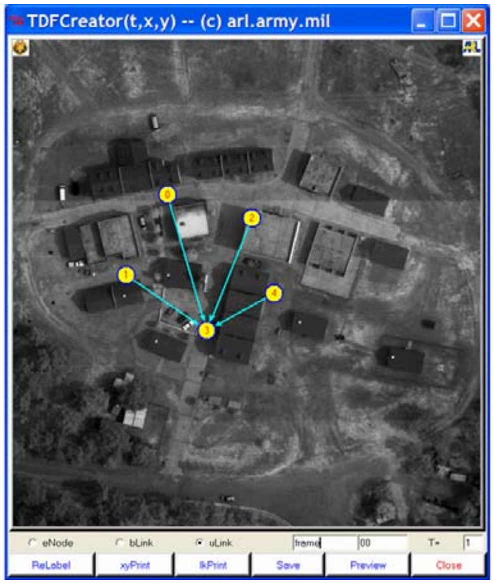

Figure 5. Creation of unidirectional links.

A unidirectional link refers to a one-way communication between two nodes. The head of the arrow points to the receiver, and the tail rests on the source node. This process also has two steps. First, the radio button labeled "uLink" is selected to start creating or deleting unidirectional links. Second, a new unidirectional link is created by moving the cursor over the receiving node, simultaneously pressing the left button and moving (dragging) the cursor to the source node, then releasing the button. As scenario files are executed at the receiving node to control inbound packets, these required actions were designed to assist the user in focusing on the receiving node.

3.4.3 Bidirectional Links

Figure 6 shows the successful creation of bidirectional links between two MANET nodes. A dual-head arrow represents a bidirectional link between two nodes, indicating that both can successfully transmit and receive messages from one another. All nodes in this figure can communicate with one another directly (one hop) or indirectly (multiple hops). A bidirectional link refers to a two-way communication between two nodes. This process also has two steps. First, the radio button labeled “bLink” is selected to start creating bidirectional links. Second, a new bidirectional link is created by moving the cursor over one of the target node, dragging the cursor to the other node. A bidirectional link is formed by a pair of unidirectional links pointing in opposite direction of each other.

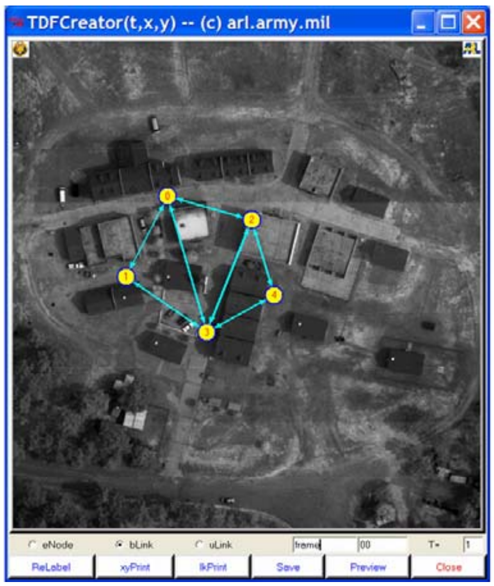

Figure 6. Creation of bidirectional links.

3.4.4 Node Relocation

Figure 7 shows the results of the relocation of the participating MANET nodes without changing the links among them. The relocation procedure has two steps: First, the radio button labeled “eNode” is selected to start the relocation process. Second, a currently displayed node is relocated by simply moving the cursor over the target node and dragging it to a new location. Existing links are automatically redrawn to reflect the change in locations.

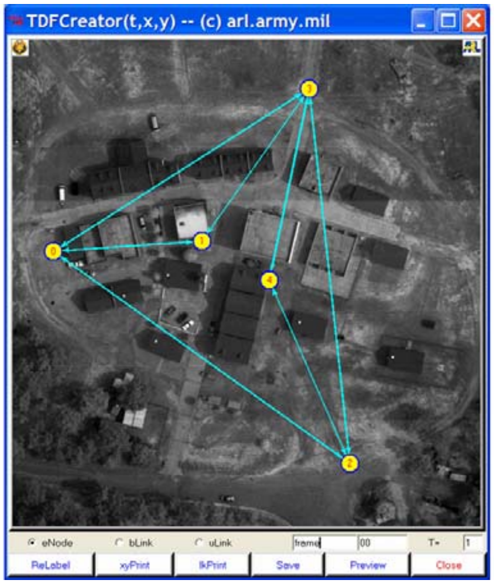

Figure 7. Relocation of nodes.

3.4.5 Topology Modification

Figure 8 shows the results of the modification to the existing MANET by adding bidirectional links between nodes 1 and 4, nodes 1 and 2, nodes 0 and 4. The procedure for adding a bidirectional link is the same as the procedure for creating new bidirectional links (figure 6).

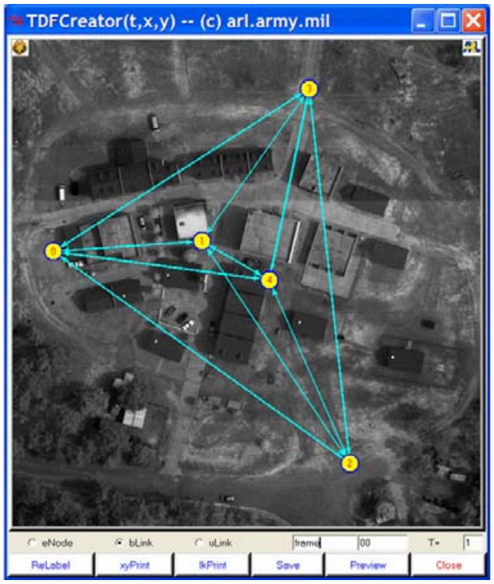

Figure 8. Addition of links.

3.4.6 Final Topology

Figure 9 shows the final results of a design by relocating the nodes to appropriate locations to form a fully connected star topology. The procedure for doing this is the same as the procedure for relocation of existing nodes as depicted in figure 7.

Figure 9. Final results.

Once the design of the star topology is completed, two additional steps are required to complete the creation of this desired topology by saving its topology definition to a file: naming a file and specifying the duration of the topology. The file name consists of a prefix and a numerical suffix, starting at zero by default. The numerical suffix is automatically incremented by one whenever the *Save* button is pressed. The time duration of the topology, specified in seconds, can be entered in a text box preceded by the label *"T="* located on the lower right corner of the screen. The default time duration is one second.

3.4.7 Action Buttons

The last row contains a set of buttons labeled *ReLabel*, *xyPrint*, *lkPrint*, *Save*, *Preview*, and *Close*. The functions of these buttons are described as follows:

- *ReLabel* – renaming the currently displayed nodes using sequential numbering method from 0 to the number of nodes – 1. This feature is used whenever the nodes are out of contiguous order due to the deletion of nodes.

- *xyPrint* – printing to the console the node locations in pixel coordinates relative to the canvas of the tool.
- *lkPrint* – printing to the console the link information using SDL.
- *Save* – saving the descriptions of the currently displayed scenario to topology definition files using the name given by the user. (The Output Files section below has more details.)
- *Preview* – visually displaying the animated sequence of the newly created topologies in the order they were saved.
- *Close* – exiting the tool.

The *TDFCreator* tool is thus easy to use because it incorporates modern interfaces familiar to users. Therefore, the learning curve is substantially minimized as its users can learn how to employ the tool within minutes.

3.5 Output Files

Each time the *Save* button is pressed, the *TDFCreator* tool generates a set of files in a directory also called *tdf*. Three types of files are saved in the *tdf* directory: topology definition *(tdf),* position information *(pos),* and composite topological information *(rgp)* files. The name *rgp* is derived from the name of the ARL engineer who creates the position data format.

3.5.1 *tdf* Files

The *tdf* files contain SDL directives. Each *tdf* file specifies a topology and its duration in seconds. This is the only data input format with which the original TSM can deal. Each directive is an SDL command to be executed at the receiving node. Only the *tdf* files are necessary for the emulation of a MANET. Appendix A has an example of the output files specifying the star topology that has been shown in figure 9.

3.5.2 *pos* Files

The ARL has added the position information in terms of pixel coordinates to each emulated topology for visualization purposes. Associated with each TDF file is a *pos* file that records the position locations of the nodes on the screen at the time the topology was saved by the user. The *pos* files together with the *tdf* files are used in the visualization and animation of dynamic topologies the MANET under test. These files also facilitate the creation, preview, and replay of emulated topologies.

3.5.3 *rgp* Files

An *rgp* file combines the topology directives and the position information. This type of file was created for the ARL-modified TSM tool.

3.5.4 Dynamic Scenario Files

The *TDFCreator* tool provides the user with a way to combine the individual static topologies into a dynamic scenario. This is a built-in feature that automatically creates a dynamic scenario file for animation purposes when the *Preview* button is pressed. The feature is also implemented in a separate tool called the "dynamic topology composer" (DTC).

A dynamic scenario file is created by combining the directives specified in a set of topology files. It can be described as an electronic flipbook containing timed sequences of static topology definition files.

This scenario file is subsequently used for animating the sequence of the newly created topologies in the order they were saved. The DTC enables its users to custom-create a composite dynamic scenario using any existing topology files.

4. Results and Discussion

The use of the *TDFCreator* tool at the ARL and at Telcordia Technologies, Inc., attests to the success of the tool and to the contribution of the ARL to the building MANET test beds and to the development of methods for visually designing and verifying desired network topologies and automatically creating topology definition files.

The *TDFCreator* tool saves substantial development time and effort of a researcher. To illustrate this claim, the following calculations are based on a hypothetical development of a single, static topology *(m=1)* of a 35-node MANET *(n=35)*.

- Number of lines of topology directives = 1225 ($mn^2 = 1 \times 35^2$)
- Number of words per line = 4 *(20 characters / 5 characters/word)*
- Number of words per topology file = 4900 *(4 x 1225)*
- Time required for manual typing = 81.67 minutes *(4900 words / 60 words per minute)*
- Time required for generating a topology file using the *TDFCreator* tool = 1 second (the estimated time required to click the left mouse button)

The calculations indicate the *minimum* amount of labor time that can be saved by using the *TDFCreator* tool, assuming that the typing was a continuous process having no errors and no interruptions and that the topology design time was the same whether the designer using pencil and paper or using point-click-drag methods.

The saved time and effort directly translates into the saving of labor costs. At a labor rate of $150/hour or $2.50/minute, each topology file generated by the tool realizes a saving of $200.00. As a typical emulation of a MANET requires at least 20 dynamic topologies, a saving of $4,000.00 can be realized. Many relevant scenarios such as this are required for a complete testing and evaluation of MANET applications.

Besides providing a way to save precious resources and shorten development time required for creating topology files, the tool enables a researcher to automatically manage the complexity of a large MANET and to visually verify the design of a dynamic scenario of a MANET. Without the *TDFCreator* tool, manually doing so would have been very difficult and time-consuming.

In summary, the *TDFCreator* tool does exactly what it is designed to do: providing MANET researchers with an automated tool and methods for systematically and quickly creating and verifying desired topologies and automatically generating topology definition files. The topology design process is accomplished through visual manipulation, inspection, and animation of generated topologies.

The desired topologies created by the *TDFCreator* tool are the products of the creativity of their creators. The locations of nodes and the links among them are subjectively established by their designers to meet the requirements of a scenario; for example, to establish a specific multi-hop network routing topology.

Having considered the constructive comments from users and observers, potential improvement to the *TDFCreator* tool in the future include the incorporation of the following possible features:

- Real-world coordinates (latitude and longitude)
- User-definable radio transmission range of each node
- User-definable velocity and a trajectory of each node
- Auto-calculated links based on estimated distances between any two nodes
- Dynamic topology derived from node movement
- Graphical representation of an individual node; e.g., soldier, vehicle, and so on.

5. Conclusions

The successful creation of the *TDFCreator* tool has boosted the building of an emulated MANET research test bed by enabling the expeditious design of desired topologies and the automatic generation of topology definition files required for the emulation of an ad hoc network.

The real impact of the *TDFCreator* tool on its users is the notable increases of their productivity and satisfaction. The *TDFCreator* tool is being used not only within ARL, but also at Telcordia Technologies, Inc.,—a premier communications research organization being a CTA member of the ARL.

Using the *TDFCreator* tool, researchers have been, for the first time, systematically and visually designing topologies and creating TDF files quickly and reliably. By using the *TDFCreator*, its users can save time, enhance efficiency, reduce complexity, and avert errors. This development has provided researchers with an unprecedented tool for creating desired MANET topologies and for effectively managing the volume and the intricacy of TDF files.

The *TDFCreator* tool benefits the researcher in at least four different ways. First, the tool provides a method for improving design accuracy through visualization of simultaneous topological links among nodes. Second, the tool significantly reduces the time required to design and develop specific MANET topologies and implementing them using the SDL because the tool automatically generates TDF files. Third, the tool assists the user in managing the size and the complexity associated with the creation of TDF files, effectively and efficiently. Fourth, the tool exempts its users from learning and writing the scenario definition language, in effect redirecting their focused efforts on the design of topologies rather than the generation of TDF directives.

6. References

1. Corson, S.; Macker, J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance: Issues and Evaluation Considerations. *The Internet Engineering Task Force (IETF) Request for Comment (RFC2501)*, January 1999. URL: http://www.ietf.org/rfc/rfc2501.txt?number=2501 (access date: 08 Sep 05)

2. Lin, Tao; Midkiff, Scott F.; Park, Jahng S. A Dynamic Topology Switch for the Emulation of Wireless Ad Hoc Networks Using a Wired Network. *Proc. 27th Annual IEEE Conference on Local Computer Network (LCN'02)*, pp. 791-798, Nov. 2002, Tampa, FL, USA

3. Science Applications International Corporation (SAIC) Research and Development Division – Co. 1, "COMTEST Scenario Generator User's Manual and Technical Reference for the Version 3.4," SAIC, 10260 Campus Point Drive, San Diego, California 92121, c/o 1410 Springhill Road, Suite 400, McLean, VA 22102, 1 October 2003

4. Zhang, Yongguang; Li, Wei. An Integrated Environment for Testing Mobile Ad-Hoc Networks. *Proc. Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, Lausanne, Switzerland, Jun 2002.

5. Macker, J. P.; Chao, W.; Weston, J. W. A Low-Cost, IP-Based Mobile Network Emulator (MNE). *Proc. Military Communications Conference, IEEE MILCOM 2003*, October 2003, pp. 481-486.

6. Karygiannis, A.; Antonakakis, E. "mLab: a Mobile Ad Hoc Network Test Bed," *1st Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing in conjunction with the IEEE International Conference in Pervasive Services 2005*, July 14, 2005, Santorini, Greece. URL: http://csrc.nist.gov/manet/SecPerU2005-Karygiannis-Antonakakis.pdf (accessed 10 September 2005).

7. Matthes, M.; Biehl, H.; Lauer, M., Drobnik, O. MASSIVE: An Emulation Environment for Mobile Ad-hoc Networks. *Proc. Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05), St. Moritz*, Switzerland, January 2005.

8. The U.S. Naval Research Laboratory, "*Nettion* (NETwork TestIng and Operational eNvironment)." URL: http://nettion.pf.itd.nrl.navy.mil/nettion_index.html (accessed 10 September 2005)

9. The Python Software Foundation, "What is Python?" URL: http:www.python.org (accessed 11 September 2005)

INTENTIONALLY LEFT BLANK.

Appendix A. A Sample of Output Files

This appendix includes the contents of the output files specifying the topology shown in figure 9 and below:

***tdf* file**

```
on 0 accept 1 inbound
on 0 accept 4 inbound
on 0 accept 2 inbound
on 0 accept 3 inbound
on 1 accept 0 inbound
on 1 accept 3 inbound
on 1 accept 2 inbound
on 1 accept 4 inbound
on 2 accept 4 inbound
on 2 accept 3 inbound
on 2 accept 1 inbound
on 2 accept 0 inbound
on 3 accept 2 inbound
on 3 accept 1 inbound
on 3 accept 0 inbound
on 3 accept 4 inbound
on 4 accept 0 inbound
on 4 accept 2 inbound
on 4 accept 3 inbound
on 4 accept 1 inbound
wait for 1 seconds
```

***pos* file**

```
0, 165, 242
1, 310, 132
2, 435, 420
3, 462, 237
4, 181, 405
wait for 1 seconds
```

***rgp* file**

```
on 0 position 165 242
on 0 accept 1 inbound
on 0 accept 4 inbound
on 0 accept 2 inbound
on 0 accept 3 inbound
on 1 accept 0 inbound
on 1 accept 3 inbound
on 1 accept 2 inbound
on 1 accept 4 inbound
on 2 accept 4 inbound
on 2 accept 3 inbound
on 2 accept 1 inbound
on 2 accept 0 inbound
on 3 accept 2 inbound
on 3 accept 1 inbound
on 3 accept 0 inbound
on 3 accept 4 inbound
on 4 accept 0 inbound
on 4 accept 2 inbound
on 4 accept 3 inbound
on 4 accept 1 inbound
on 1 position 310 132
on 2 position 435 420
on 3 position 462 237
on 4 position 181 405
wait for 1 seconds
```

INTENTIONALLY LEFT BLANK.

Appendix B. Acronyms and Abbreviations

COMTEST™	A suite of software tools designed for the development of military communications systems developed by Science Applications International Corporation (SAIC)
CSG	Communications Scenario Generator – part of the COMTEST™ software suite.
Iptables	An open-source packet-filtering and mangling software tool available in the Linux operating systems. URL: http://www.netfilter.org/
MANET	*Mobile ad hoc network – an autonomous system of collaborative computing platforms that can function without relying on an establish network infrastructure.* *URL: http://www.ietf.org/rfc/rfc2501.txt?number=2501*
POS	Position information in pixel coordinates relative to the canvas area of the tool
SDL	A scenario definition language adopted by ARL and Telcordia Technologies, Inc., to define an emulated topology of a MANET
TDF	Topology definition file format defined by Telcordia Technologies, Inc., It is a text file containing sequences of topological directives expressed in the scenario definition language (SDL)
TDFCreator	A graphical tool that enables the end-user to visually design desired topologies and to automatically generate TDF files required for the emulation of a MANET.
RGP	The topology data format defined by Richard Gopaul of Army Research Laboratory, Adelphi, Maryland

INTENTIONALLY LEFT BLANK.

Distribution

ADMNSTR
DEFNS TECHL INFO CTR
ATTN DTIC-OCP (ELECTRONIC COPY)
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

DARPA
ATTN IXO S WELBY
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

OFC OF THE SECY OF DEFNS
ATTN ODDRE (R&AT)
THE PENTAGON
WASHINGTON DC 20301-3080

US ARMY TRADOC
BATTLE LAB INTEGRATION & TECHL DIRCTRT
ATTN ATCD-B
10 WHISTLER LANE
FT MONROE VA 23651-5850

SMC/GPA
2420 VELA WAY STE 1866
EL SEGUNDO CA 90245-4659

US ARMY ARDEC
ATTN AMSTA-AR-TD
BLDG 1
PICATINNY ARSENAL NJ 07806-5000

COMMANDING GENERAL
US ARMY AVN & MIS CMND
ATTN AMSAM-RD W C MCCORKLE
REDSTONE ARSENAL AL 35898-5000

US ARMY INFO SYS ENGRG CMND
ATTN AMSEL-IE-TD F JENIA
FT HUACHUCA AZ 85613-5300

US ARMY SIMULATION TRAIN & INSTRMNTN CMND
ATTN AMSTI-CG M MACEDONIA
12350 RESEARCH PARKWAY
ORLANDO FL 32826-3726

US GOVERNMENT PRINT OFF
DEPOSITORY RECEIVING SECTION
ATTN MAIL STOP IDAD J TATE
732 NORTH CAPITOL ST., NW
WASHINGTON DC 20402

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-OK-TP TECHL LIB T LANDFRIED (2 COPIES)
ABERDEEN PROVING GROUND MD 21005-5066

DIRECTOR
US ARMY RSRCH LAB
ATTN AMSRD-ARL-RO-EN W D BACH
PO BOX 12211
RESEARCH TRIANGLE PARK NC 27709

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-CN B NGUYEN (2 COPIES)
ATTN AMSRD-ARL-CI-CN G RACINE (2 COPIES)
ATTN AMSRD-ARL-CI-OK-T TECHL PUB (2 COPIES)
ATTN AMSRD-ARL-CI-OK-TL TECHL LIB (2 COPIES)
ATTN AMSRD-ARL-D J M MILLER
ATTN AMSRD-ARL-IS-CB P BUDULAS (2 COPIES)
ATTN IMNE-ALC-IMS MAIL & RECORDS MGMT
ADELPHI MD 20783-1197

INTENTIONALLY LEFT BLANK.